

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-242967

(43)Date of publication of application : 02.09.1994

(51)Int.Cl.

G06F 9/46

(21)Application number : 05-026633

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 16.02.1993

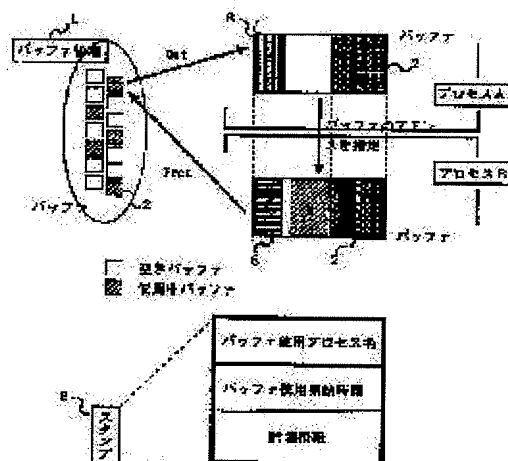
(72)Inventor : SHIMIZU KEIICHI

## (54) BUFFER OPERATION METHOD AND BUFFER MANAGEMENT DEVICE

## (57)Abstract:

PURPOSE: To detect stagnation on time by providing a means for specifying a process name which caused an error and a fault when a buffer is stagnated inside a process by the error and the fault of the process.

CONSTITUTION: The process A stamps the process name, present time and other detail information in a stamp area 8 after acquiring the buffer 2. Then, the process A performs message communication to the process B, the address of the buffer 2 is presented and the buffer 2 transfers a using right to the process B. When the buffer 2 is acquired, the process B stamps the process name, the present time and the other detail information in the stamp area 8. At this time, stamped contents stamped by the process A are erased and the stamped contents stamped by the process B are substituted. When the buffer 2 is stagnated by the error of a certain process, a stagnating process name can be defined by analyzing the buffer 2 in use managed by buffer management 1.



(11)特許出願公開番号

特開平6-242967

(43)公開日 平成6年(1994)9月2日

### 技術表示箇所

3 4 0 B 8120-5B

審査請求 未請求 請求項の数 4 OL (全 10 頁)

(74)代理人 弁理士 吉田 研二 (外2名)

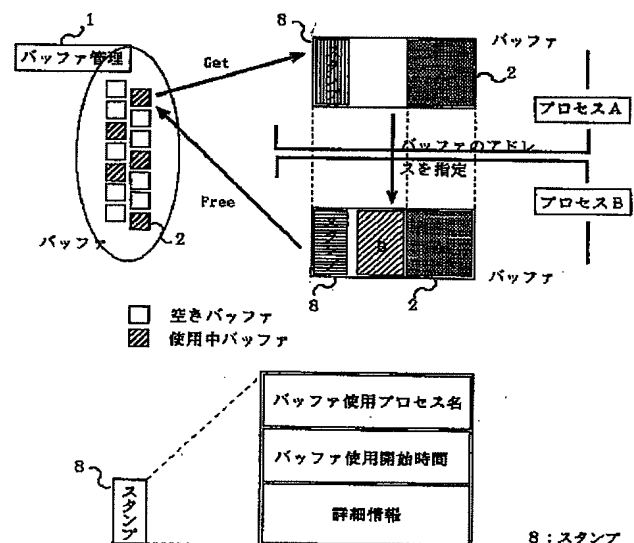
(54)【発明の名称】 バッファ操作方法及びバッファ管理装置

(57) 【要約】

【目的】 複数のプロセスが共通にアクセス可能なバッファを引き継いで処理を行うデータ処理システムにおいて、プロセスのミスや障害によって、当該のプロセス内部でバッファが停滞した場合に、ミスのあるプロセスを特定し、さらに原因の解析を容易にするバッファ操作方法およびバッファ操作装置を提供する。

【構成】 バッファの使用権を取得した際、使用を開始するプロセスの名前、現時刻、他詳細情報のうち、少なくとも使用を開始するプロセスの名前をスタンプする方法と所定のスタンプ領域を有している。

【効果】 スタンプ領域からの情報により、バッファの停滞原因の早期発見、解析が容易になり、プロセスのミスの早期低減が可能なバッファ操作方法およびバッファ操作装置の提供を可能とする。



1

## 【特許請求の範囲】

【請求項1】 複数のプロセスが共通にアクセス可能なバッファを有するデータ処理システムにおけるバッファ操作方法において、プロセスがバッファの使用権を取得後、使用権を取得したプロセスの名前が前記データ処理システム内の所定のスタンプ領域にスタンプされることを特徴とするバッファ操作方法。

【請求項2】 複数のプロセスが共通にアクセス可能なバッファを有するデータ処理システムにおけるバッファ操作方法において、プロセスがバッファの使用権を取得後、使用権を取得したプロセスの名前と現時刻とが前記データ処理システム内の所定のスタンプ領域にスタンプされることを特徴とするバッファ操作方法。

【請求項3】 複数のプロセスが共通にアクセス可能なバッファを有するデータ処理システムにおけるバッファ操作装置において、プロセスが使用するバッファの使用時間を計測する手段と、この計測した使用時間が一定時間を越えた場合、警告信号を発生する警告手段とを有することを特徴とするバッファ管理装置。

【請求項4】 CTRON (Communication TRON) によるプロセス間メッセージ通信に係るバッファ操作方法において、プロセスがバッファの使用権を取得後、使用権を取得したプロセスの名前を所定のスタンプ領域にスタンプすることを特徴とするバッファ操作方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、コンピュータ等によるデータ処理システムにおいて、特に、複数のプロセスが共通にアクセス可能なバッファを引き継いで処理を行うデータ処理システムに関するものである。

## 【0002】

【従来の技術】 図8は、例えば、社団法人トロン協会が出版している「原典CTRON大系通信制御インタフェース 各層共通・基本通信制御編」1989年7月10日頁20～21に記載されている、プロセス間で共通にアクセス可能なバッファを有する従来のバッファ操作方式を説明する図である。

【0003】 この図におけるバッファ操作方式においては、一つのバッファを二つのプロセスが共通に使用することが想定されている。バッファ管理プロセス1（以下、単にバッファ管理1と言う）は、複数のプロセスによって共通にアクセス可能なバッファ2を一括して管理している。次に図8に基づいて、動作を説明する。プロセスAは、バッファ管理1にバッファ2の一つの使用権を取得（以下、単に取得と言う）するための要求をする。バッファ管理1は未使用のバッファがあれば、そのメモリ上の一つのアドレスをプロセスAに提示し、前記バッファが使用中である旨記録する。使用中と記録された前記アドレスのバッファ2を取得したプロセスAが、そのバッファを取得（GET）したことになる。その

2

後、バッファ2はプロセスAによる操作が行われる。ここでプロセスAによる操作とは、例えば、プロセスAによりバッファ2に情報Aが書き込まれる操作である。

【0004】 次に、プロセスAからプロセスBに、メッセージ通信を行い、バッファ2の使用権がプロセスAからプロセスBへ移る。その後、バッファ2はプロセスBによる操作が行われる。ここでプロセスBによる操作とは、例えば、プロセスBによりバッファ2に情報Bが書き込まれる操作である。また、この時バッファ管理1におけるバッファ2の状態は使用中のまま変化しない。

【0005】 バッファ2のプロセスBによる操作後、プロセスBはバッファ2をバッファ管理1へ返却（FREE）する。バッファ管理1は、解放されたバッファ2の状態を未使用状態に設定する。

【0006】 図9は任意の一個のバッファが、2個以上のプロセスにより連鎖的に引き継がれる場合の、前記プロセスの操作を説明するための図である。次に動作について説明する。プロセスAがバッファ2をバッファ管理1より取得した後、バッファ2はプロセスB～Eまで引き継がれ、各プロセスでバッファ操作が行われる。最終的にプロセスEがバッファを返却する。

【0007】 図10に各プロセスにおけるバッファの操作のフローチャートを示す。この図は、図9で示した共通のバッファ2を使用する複数のプロセスのバッファ操作のフローチャートである。プロセスAが、まずバッファ管理1よりバッファ2を取得する。プロセスAはバッファ2に所定の操作をし、次のプロセスBへメッセージを送信し、バッファ2を渡す。その後プロセスAは、必要ならば再びバッファ管理1よりバッファ2を取得する。プロセスAよりバッファ2を引き継いだプロセスBは、バッファ2に所定の操作を加え、次のプロセスへバッファ2を渡す。この後プロセスBは必要ならば、再びプロセスAからのメッセージ受信待ちとなる。

【0008】 当該のバッファ2を最後に使用するプロセスEは、メッセージ受信時にバッファを受取り、このバッファへの操作終了後、バッファ2をバッファ管理1へ返却する。この後プロセスEは必要ならば、再びメッセージ受信待ちとなる。

## 【0009】

【発明が解決しようとする課題】 上記のように、同一のバッファを複数のプロセスがアクセスし、アクセスが多段におよぶようなデータ処理において、プロセスのミス（バグ）等による障害により、当該のプロセス内部でバッファが停滞、またはプロセスが解放されないまま浮いてしまった場合、ミスや障害のあるプロセスを特定するのが困難であるという問題点があった。以下にその具体例を示す。

【0010】 図11は、プロセスDのミスにより、バッファ2a～2eがプロセスD内部で停滞してしまった例を示している。この時、バッファ2a～2eはプロセス

3

Eに引き継がれないため、プロセスDは開放されず、浮いた状態のままとなる。このように、図9で示めすような正常なデータ処理情報は不可能となり、バッファが途中で停滞したものとなる。

【0011】または、プロセスDの他のミス、例えばプロセスDが必要個数以上のバッファ2の取得を要求するミスが生じ、未使用のバッファ2が無くなるまで、バッファ2の取得を要求した場合、未使用のバッファ2が足りなくなり、プロセスAがバッファ2を取得することができず、プロセスAは、処理を停滞させ浮いたままの状態となり、目的とするデータ処理を実行することが不可能である。この時バッファ管理1におけるバッファ2の使用中的アドレスは特定できるが、そのアドレスのバッファがどのプロセスで使用されているかは不明であり、バッファを使用しているプロセス名は判別できない。また、このようなバッファ操作方式では、最終的に未使用のバッファが全く無くなるまで、ミスが検出されないといった問題点がある。

【0012】本発明は上記のような問題点を解決するためになされたもので、プロセスのミスや障害によって、バッファがプロセス内部に停滞してしまった場合、ミスや障害を起こしたプロセス名を特定する手段を提供し、停滞したことをオンタイムで検出するバッファ操作方法、及びバッファ操作装置の実現を目的とする。

【0013】

【課題を解決するための手段】本発明は、複数のプロセスが共通にアクセス可能なバッファを有するデータ処理システムにおけるバッファ操作方法において、プロセスがバッファの使用権を取得後、使用権を取得したプロセスの名前が前記データ処理システム内の所定のスタンプ領域にスタンプする方法を有している。

【0014】または、複数のプロセスが共通にアクセス可能なバッファを有するデータ処理システムにおけるバッファ操作方法において、プロセスがバッファの使用権を取得後、使用権を取得したプロセスの名前と現時刻とが前記データ処理システム内の所定のスタンプ領域にスタンプする方法を有している。

【0015】または、複数のプロセスが共通にアクセス可能なバッファを有するデータ処理システムにおけるバッファ操作装置において、プロセスが使用するバッファの使用時間を計測する手段と、この計測した使用時間が一定時間を越えた場合、警告信号を発生する警告手段とを有している。

【0016】あるいは、CTRON (Communication TRON) によるプロセス間メッセージ通信に係るバッファ操作方法において、プロセスがバッファの使用権を取得後、使用権を取得したプロセスの名前を所定のスタンプ領域にスタンプする方法を有している。

【0017】

【作用】本発明において、所定のスタンプ領域に、バッ

4

ファを取得したプロセスの名前をスタンプすることにより、あるプロセスのミスによってバッファの空きがなくなってしまう場合、スタンプ領域を解析することにより、ミスのあるプロセス名を一意に特定できる。

【0018】または、スタンプ領域にプロセスの名前の他、現時刻もスタンプすることにより、スタンプされる時刻は、バッファの使用開始時間であり、この使用開始時間とデータ処理システムの現時刻との差をとり、この差が一定以上となった時、バッファが停滞していると判断させ、データ処理システムに対して、バッファの停滞の警告信号を発生させ、さらに、その警告により、データ処理を停止させる等が可能であり、バッファの停滞の早期発見、原因解析が容易になり、プロセスのミスの早期低減を容易にする。

【0019】または、プロセスが使用するバッファの使用時間を計測する手段と、この使用時間が一定時間を越えた場合、停滞の警告信号を発生させる警告発生手段を備え、データ処理システムに対して、プロセス内のバッファの停滞の警告信号を発生させ、さらに、その警告により、データ処理を停止させる等が可能であり、バッファ停滞の早期発見、原因解析が容易になり、プロセスのミスの早期低減を容易とする。

【0020】あるいは、CTRON (Communication TRON) によるプロセス間メッセージ通信に係るバッファ操作方法において、プロセスがバッファの使用権を取得した後、使用権を取得したプロセスの名前を、所定のスタンプ領域スタンプすることにより、スタンプ領域を解析することにより、あるプロセスのミスによってバッファの空きがなくなってしまう場合、スタンプ領域を解析することにより、ミスのあるプロセス名を一意に特定できる。

【0021】

【実施例】

実施例1. 図1は本発明の一実施例を示すものであり、1~4は従来技術で説明したものと全く同一のものである。8はバッファ中に設けられたスタンプ領域を示しており、このスタンプ領域は、本実施例において特徴的であり、バッファを使用するプロセス名、バッファの使用開始時刻である現時刻、プロセスのミスを解析するための他詳細情報からなる。

【0022】次に動作について説明する。図1においてプロセスAが、バッファ管理1にバッファ2を一つ要求する。バッファ管理1は未使用のバッファ2があれば、そのバッファ2をプロセスAに提示後、そのバッファ2を使用中と認識する。以上より、プロセスAは、そのバッファ2を取得(GET)する。

【0023】プロセスAはバッファ取得後、本実施例において特徴的であるスタンプ領域に、プロセス名、現時刻、他詳細情報をスタンプする。ここで、詳細情報はプロセスのミスが生じた際、できるかぎり容易にミスの箇

5

所を特定できるよう、プロセスが独自に定義したもので、例えばプロセスで用いる特定ファイルのデータ等である。つぎに、プロセスAはバッファ2に対して操作を行い、例えば情報Aを書き込んだとする。この後、プロセスAはプロセスBにメッセージ通信を行い、バッファ2のアドレスがプロセスBに提示される。この時、バッファ2は使用権をプロセスBに移行する。この後プロセスAは必要ならば、再びバッファの取得を繰り返す。

【0024】プロセスBはバッファ2をプロセスAから取得すると、プロセスBは、本実施例において特徴的であるスタンプ領域に、プロセス名、現時刻、他詳細情報をスタンプする。このとき、プロセスAが押したスタンプ内容は消去され、プロセスBが押したスタンプ内容が上書き（以下、単に上書きと言う）される。つぎに、プロセスBは同一のバッファ2に対して操作を行い、例えば情報Bを書き込む。この後、プロセスBは、バッファ2のデータ処理を必要とし無くなったとき、プロセスBはバッファ2をバッファ管理1へ返却（FREE）する。ここで、バッファ管理1は、解放されたバッファ2の状態を未使用状態に設定する。この後プロセスBは必要ならば、再びバッファの取得を繰り返す。

【0025】以上プロセスがバッファ2を取得後、プロセスがバッファ2にスタンプする際の動作説明を行った。さらに、図9にプロセスが2個以上となった場合を示す。また、この場合の各プロセスのバッファ操作のフローチャートを図2に示す。以下、図2、図9による動作説明を行う。

【0026】バッファ2がプロセスAからプロセスBに引き継がれる場合の動作については、上記において説明を行った。さらに、同様の動作がプロセスBからプロセスC、プロセスCからプロセスD、プロセスDからプロセスEと順次行われる。この時、バッファ2を最後に使用するプロセスEは、メッセージ受信時にバッファ2を受け取り、まずバッファ2にスタンプを上書きする。バッファ操作終了後、バッファ2をバッファ管理1へ返却する。この後プロセスEは必要ならば、再びメッセージ受信待ちとなる。

【0027】ここで、あるプロセスのミスによって、一定時間を越えて一つのプロセス内にバッファ2が停滞し、使用可能なバッファがなくなってしまう場合、バッファ管理1はバッファ2の取得要求に応じられなくなり、何らかのミスが発覚する。このとき、バッファ管理1が管理する使用中のバッファ2を解析することにより、停滞したプロセス名を確定できる。この確定は、スタンプ領域にプロセスの名前をスタンプすることにより可能である。

【0028】本実施例においては、プロセス中でのバッファの使用状態の詳細情報をバッファの属性として、スタンプしているので、プロセスのミスの解析がより容易になる。さらに、バッファの使用開始時間である現時刻

6

をスタンプしている。この使用開始時間とデータ処理システムの現時刻との差をとり、この差が一定以上となった時、バッファが停滞していると判断し、データ処理システムに対して、バッファの停滞警告信号をオンタイムで発生させる。さらに、その警告信号により、データ処理を停止させる等が可能であり、バッファの停滞の早期発見、原因解析が容易になり、プロセスのミスの早期低減を容易にする。

【0029】実施例2. 上記実施例1ではプロセス領域でスタンプ操作を管理していたが、本実施例では、プロセス領域以外のバッファ管理領域でスタンプ操作を管理する。このバッファ管理領域は、スタンプ操作に係るスタンプ領域を有し、このスタンプ領域は各バッファ毎に対応するアドレスが各々に割り付けられている。

【0030】図3は本実施例における、スタンプ方法のフローチャートを示し、ある任意のプロセスにバッファを引き継ぎ、引継ぎ後、前記任意のプロセスが、前記バッファをプロセスBに渡すまでのフローチャートを示す。次に動作説明を行う。初めに、ある任意のプロセスが、他のプロセスよりバッファを引き継ぐ時、前記任意のプロセスは、引き継ぎのメッセージを受ける。前記バッファを引き継ぐようメッセージを受けた前記任意プロセスは、バッファ管理にスタンプを要求する。前記任意プロセスより、スタンプの要求を受信したバッファ管理は、まずデータ処理システムより現時刻を取得し、前記プロセスから提示されたプロセス名、プロセスのミスを解析するための他詳細情報とともに前記バッファのスタンプ領域へ、その内容を上書きする。

【0031】図4は本実施例における、バッファ管理領域の一例を図示したものである。バッファ管理領域は、バッファの使用状況を表すフラグの他に、上記実施例1にて定義したスタンプ情報、およびバッファの使用時間をカウントするタイムカウント値が含まれている。このように、プロセス領域以外のバッファ管理領域でスタンプ操作を管理することにより、スタンプの管理が容易となるだけでなく、バッファを操作するプロセス群の処理が軽減される。

【0032】実施例3. 図5は本実施例を説明するためのものであり、実施例1で示したスタンプ領域をCTRLONの通信制御インタフェースで使用するバッファに適用したものである。CTRLON通信によるバッファはインフォメーションブロックと呼ばれ、このインフォメーションブロック9は、バッファ2のアドレスを設定するdata-locationと、プロセスによるバッファ2へのデータ書き込み開始位置を設定するstart-byte-positionと、複数のバッファ内の全データ長を設定するdata-lengthと、ユーザプロセス情報の書き込み開始位置を設定するstart-byte-positionを有する。本実施例では、バッファ2のbuffer-block-dat

a-lengthの後からstart-byte-positionで示されるアドレスまでの間の空き領域にスタンプ情報を設定する。

【0033】このような設定は、スタンプ領域を実施例2で説明したように、バッファ管理領域内に設けることも可能である。

【0034】実施例4. 図4は本実施例を示すものであり、特に、バッファ管理領域内に、バッファの使用時間を計測するためのタイムカウント値を有することを特徴とする。このタイムカウント値は、バッファ毎に一对一に対応しており、バッファ使用開始とともに、バッファ管理により一定周期でインクリメントされる。

【0035】次にバッファ管理の動作を記述する。図6、図7はバッファ停滯を自動的に検出する際のバッファ管理のフローチャートを示したものである。図6はユーザからの要求によって動作する機能部を示しており、図7は周期的に動作する機能部を示している。

【0036】次にその動作を説明する。周期時間をカウントする周期タイマの起動は、データ処理システム内のバッファ管理の起動と同時に進行。バッファ管理は、バッファ取得の要求を受信すると、使用するバッファを決定し、そのバッファのスタンプ領域にスタンプを押す。このとき、タイムカウント値は初期値に設定後、インクリメントを開始する。このタイムカウント値が、一定時間（この一定時間はデータ処理システムに依存する）を越えた場合、データ処理システムに警告信号を発生させる。さらにこの警告信号により、プロセスのミスや障害によってバッファが停滯してしまったと判断し、データ処理システムを停止させる等が可能である。

【0037】また、プロセスにミスが無い場合は、プロセスによるバッファの操作を正常に終了した後、バッファ管理が、バッファの開放を受信すると、バッファ管理は、タイムカウント値を初期値に設定し、スタンプ情報をクリアする。

【0038】以上のように、プロセスが使用するバッファの使用時間を計測する手段を備えることにより、一定時間を越えて一つのプロセス内にバッファが停滯していることを検出し、システムに対してオンタイムで警告信号を発生することが可能である。

【0039】上記実施例1～4においては、バッファを使用するユーザがプロセスであったが、ユーザはタスクであっても一つのサブルーチンであってもかまわない。

【0040】

【発明の効果】以上説明したように、本発明において、スタンプ領域にプロセスの名前だけがスタンプされるバッファ操作方法とした場合、プロセスのミスでバッファが停滯し、最終的に未使用のバッファがなくなった時、初めてミスが検出されるが、停滯の原因となったプロセスの名前は、停滯したバッファのスタンプ領域に記載されており、バッファの解析によりプロセスの名前の特定が

行われ、バッファの停滯原因の発見、解析が容易となり、プロセスのミスの低減を容易とするバッファ操作方法を可能とする。

【0041】または、スタンプ領域にプロセスの名前、バッファの使用開始時刻である現時刻をスタンプした場合、このバッファの使用開始時刻とデータ情報システムの現時刻との時間差をとり、この時間差が一定時間を越えて一つのプロセス内で継続して使用されているかどうかを判断することが可能である。さらにこのバッファが、ある一定時間を越えて継続して使用されていることを検出した場合、データ処理システムに対して、バッファの停滯の警告信号をオンタイムで発生させたり、データ処理を停止させる等が可能であり、バッファの停滯原因の早期発見、解析が容易になり、プロセスのミスの早期低減を可能とするバッファ操作方法を可能とする。

【0042】または、プロセスが使用するバッファの使用時間を計測する手段と、この使用時間が一定時間を越えた場合、システムに対してバッファ停滯の警告信号をオンタイムで発生させる警告手段を備えることが可能であり、この警告によりプロセスのミスの早期発見、解析が容易になり、プロセスのミスの早期低減を可能とするバッファ管理装置を可能とする。

【0043】あるいは、CTRON (Communication TRON) によるプロセス間メッセージ通信に係るバッファ操作方法において、プロセスがバッファの使用権を取得した後、使用権を取得したプロセスの名前を所定のスタンプ領域にスタンプするためのスタンプ領域を有することにより、あるプロセスにミスが生じた場合、スタンプ領域を解析することによりどのプロセスのミスかを一意に特定でき、バッファの停滯原因の発見、解析が容易となり、プロセスのミスの低減を容易とするバッファ操作方法を可能とする。

【図面の簡単な説明】

【図1】本発明に係るバッファ操作方法の、好適な実施例を説明する図である。特にスタンプ領域を設定したバッファ操作方法を説明する図である。

【図2】本発明の実施例1に係る、バッファを使用する各プロセスの、バッファ操作を説明するフローチャートである。

【図3】本発明の実施例2に係る、バッファを使用する各プロセスおよびバッファ管理の、バッファ操作を説明するフローチャートである。

【図4】本発明の実施例2及び、実施例4に係る、バッファ管理領域内部の情報配置を説明する図である。

【図5】本発明の実施例3に係る、CTRON通信制御インタフェースの、インフォメーションブロックおよびバッファの構成を説明する図である。

【図6】本発明の実施例4に係る、停滯を自動的に検出する機構を備えたバッファ操作を説明するフローチャートである。

9

【図7】本発明の実施例4に係る、停滞を自動的に検出する機構を備えたバッファ操作を説明するフローチャートである。

【図8】従来のバッファ操作方式を説明する図である。

【図9】バッファが複数のプロセスに引き継がれて処理される、データ処理システムを説明する図である。

【図10】従来のバッファ操作方式による、バッファ操作を説明するフローチャートである。

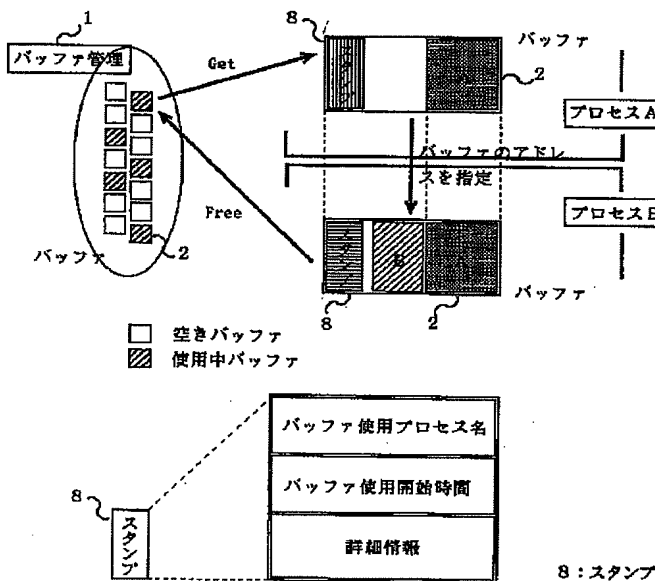
10

【図11】従来のバッファ操作方式による問題点を説明する図である。

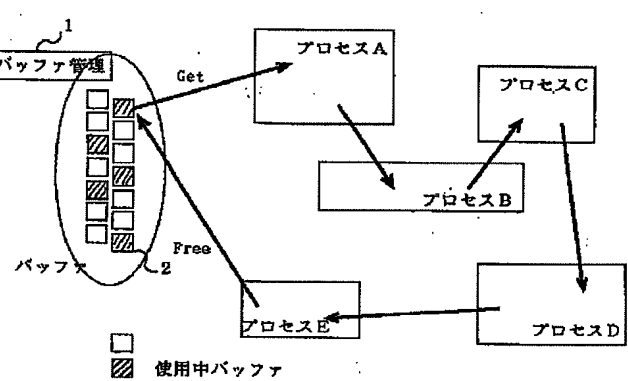
【符号の説明】

- 1 バッファ管理プロセス
- 2、2a～2e バッファ
- 8 スタンプ
- 9 インフォメーションブロック

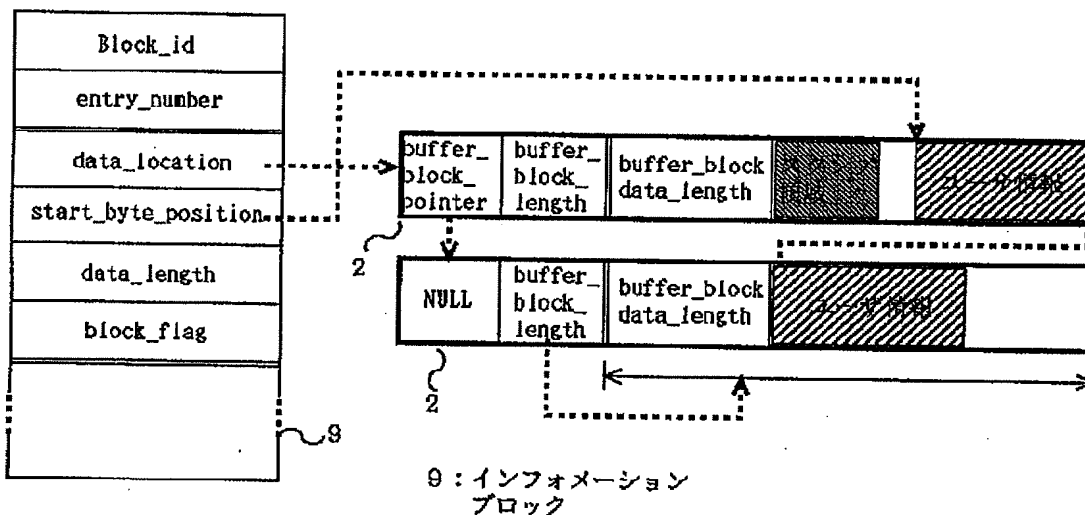
【図1】



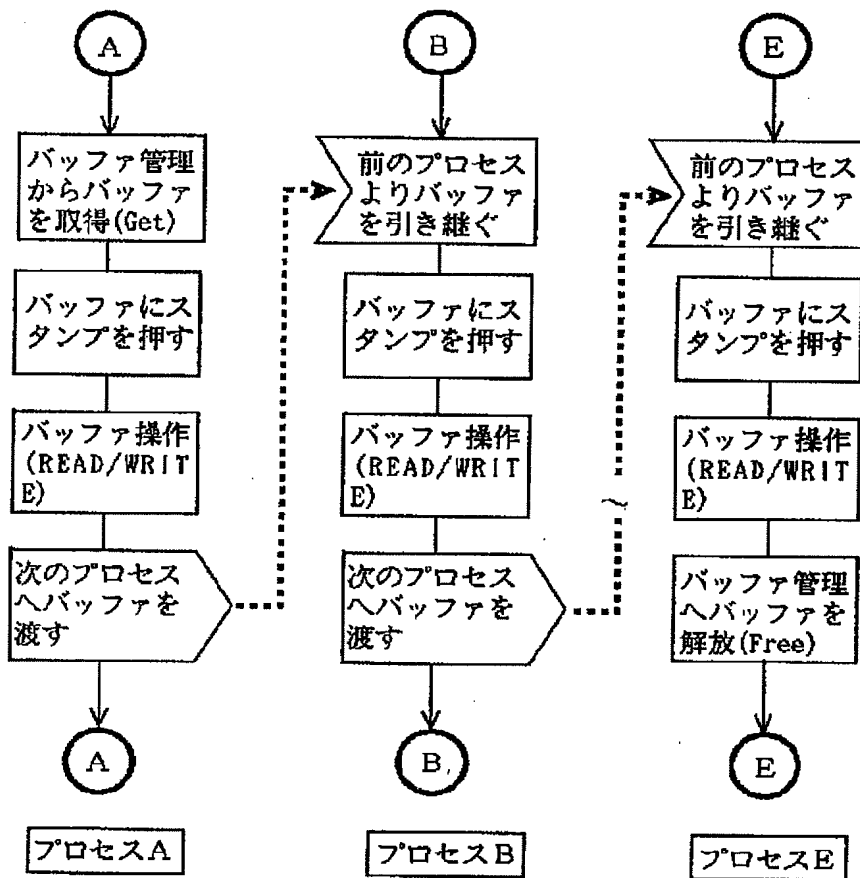
【図9】



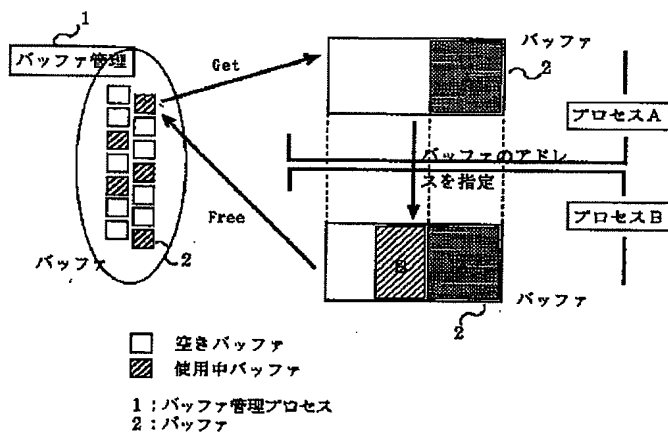
【図5】



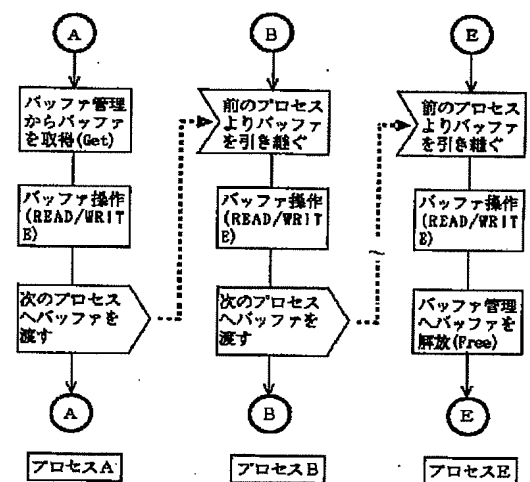
【図2】



【図8】

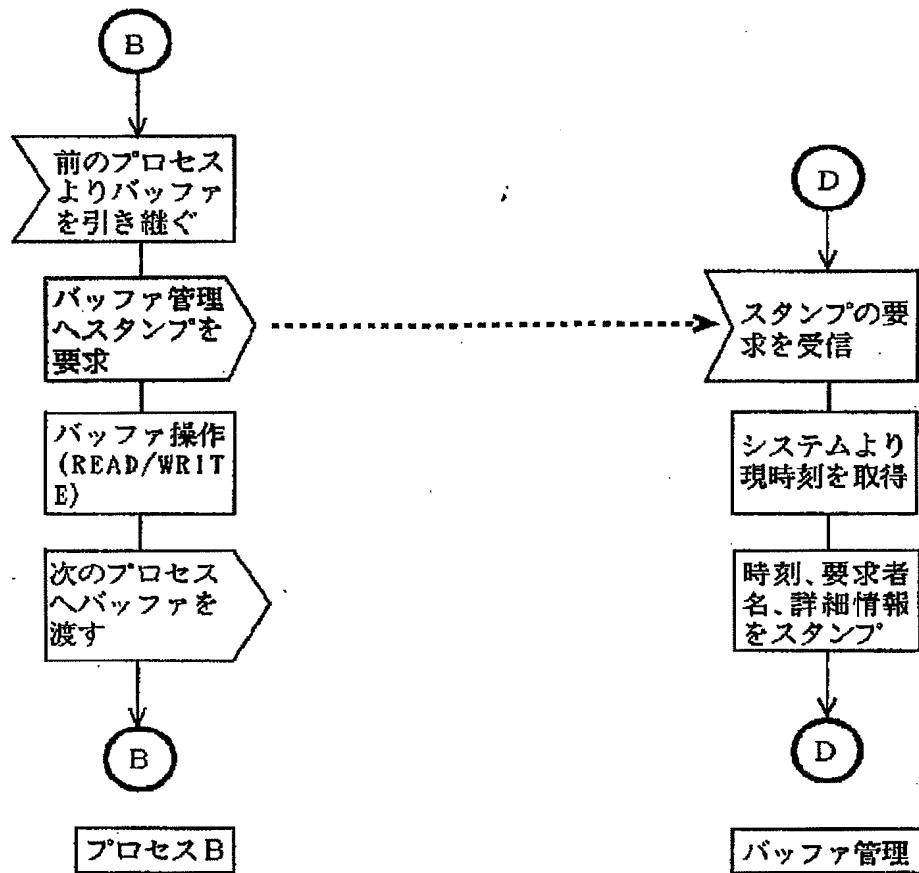


【図10】

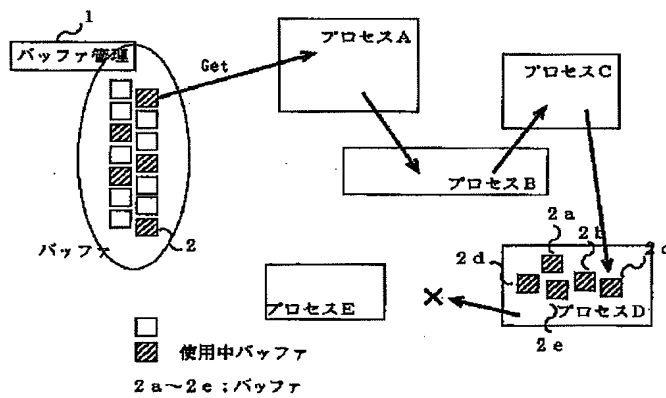




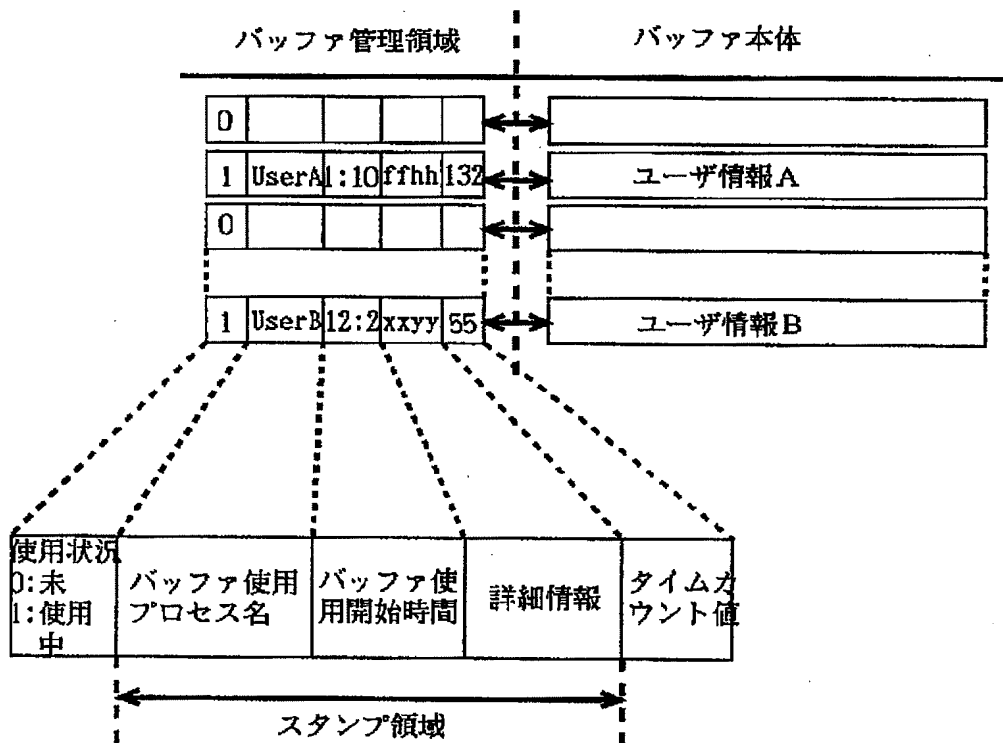
【図3】



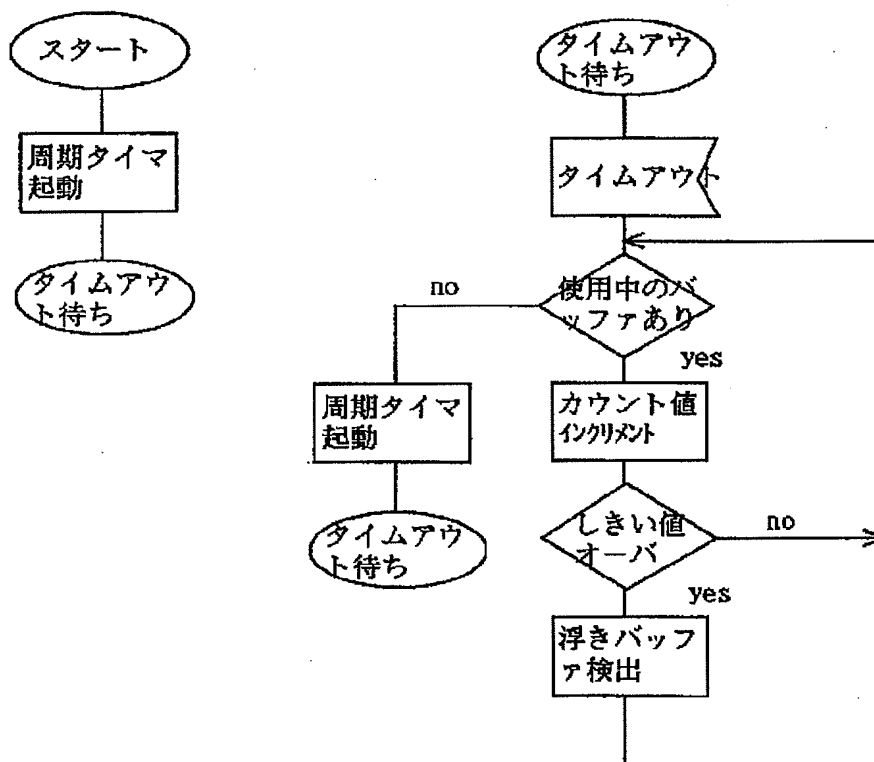
【図11】



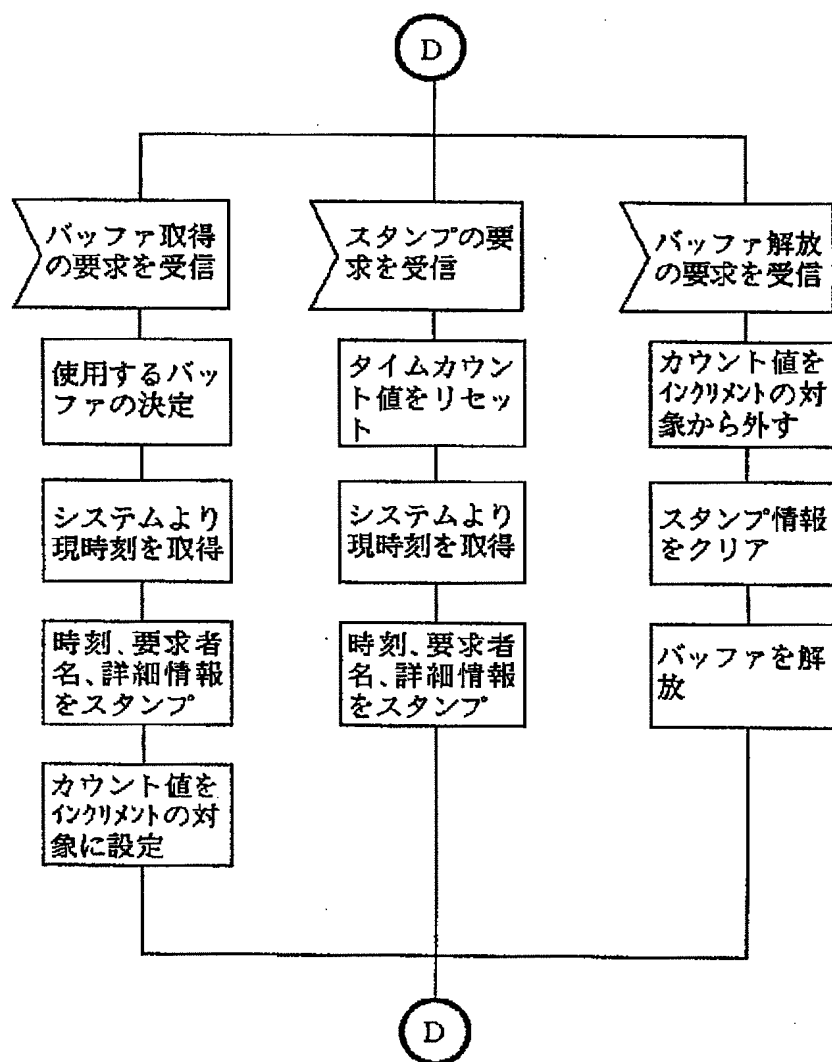
【図4】



【図7】



【図6】



## 【手続補正書】

【提出日】平成5年8月17日

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0032

【補正方法】変更

## 【補正内容】

【0032】実施例3. 図5は本実施例を説明するためのものであり、実施例1で示したスタンプ領域をCTRONの通信制御インタフェースで使用するバッファに適用したものである。CTRON通信によるバッファはイ

ンフォメーションブロックと呼ばれ、このインフォメーションブロック9は、バッファ2のアドレスを設定するdata-locationと、プロセスによるバッファ2へのデータ書き込み開始位置を設定するstart-byte-positionと、複数のバッファ内の全データ長を設定するdata-lengthを有する。本実施例では、バッファ2のbuffer-block-data-lengthの後からstart-byte-positionで示されるアドレスまでの間の空き領域にスタンプ情報を設定する。